### 15,05,2018 Коллекции Generic — предки и наследники 5

Теперь сделаем очень простой пример, в котором заполним коллекции из двух классов-наследников и сделаем метод, который печатает поле **name** класса **BasicClass**.

возможности generic. Создадим очень простой класс с одним полем — **BasicClass**

====================================================================

package edu.javacourse.collection;

public class BasicClass

{

    private String name;

    public BasicClass(String name) {

        this.name = name;

    }

    public String getName() {

        return name;

    }

}

package edu.javacourse.collection;

public class MyClass1 extends BasicClass

{

    public MyClass1(String name) {

        super(name);

    }

}

package edu.javacourse.collection;

public class MyClass2 extends BasicClass

{

    public MyClass2(String name) {

        super(name);

    }

}

 А хочется научить компилятор понимать, что мы хотим передавать в метод **printCollection** список, элементы которого являются НАСЛЕДНИКАМИ **BasicClass**. И generic предлагает такое решение. Смотрим новый вариант и сосредоточим наше внимание опять на объявление метода **printCollection**.

Наш метод принимает список, элементы которого являются наследниками класса **BasicClass**. То, что нам надо. С одной стороны мы можем передавать наследников (до любого колена), с другой — у нас есть ограничение. Я не смогу передать в метод **printCollection** список с элементами типа **String** или **Integer**. Более точные правила, более жесткие контракты — меньше ошибок в большом и сложном проекте. Есть контакт.

package edu.javacourse.collection;

import java.util.ArrayList;

import java.util.List;

public class Main

{

    public static void main(String[] args) {

        List<MyClass1> list1 = new ArrayList<>();

        list1.add(new MyClass1("Василий 1"));

        list1.add(new MyClass1("Павел 1"));

        list1.add(new MyClass1("Андрей 1"));

        list1.add(new MyClass1("Петр 1"));

        list1.add(new MyClass1("Анжелика 1"));

        printCollection("Коллеция 1", list1);

        List<MyClass2> list2 = new ArrayList<>();

        list2.add(new MyClass2("Василий 2"));

        list2.add(new MyClass2("Павел 2"));

        list2.add(new MyClass2("Андрей 2"));

        list2.add(new MyClass2("Петр 2"));

        list2.add(new MyClass2("Анжелика 2"));

        printCollection("Коллекция 2", list2);

    }

    private static void printCollection(String title, List<? extends BasicClass> list) {

        System.out.println(title);

        for (BasicClass bc : list) {

            System.out.println("Item:" + bc.getName());

        }

        System.out.println();

    }

}

Но существует обратный вариант ограничения — когда вы сможете передавать с коллекцию только те классы, которые являются ПРЕДКАМИ

package edu.javacourse.collection;

import java.util.ArrayList;

import java.util.List;

public class Main

{

    public static void main(String[] args) {

        List<BasicClass> list1 = new ArrayList<>();

        list1.add(new MyClass1("Василий 1"));

        list1.add(new MyClass1("Павел 1"));

        list1.add(new MyClass1("Андрей 1"));

        list1.add(new MyClass1("Петр 1"));

        list1.add(new MyClass1("Анжелика 1"));

        printCollection("Коллеция 1", list1);

        List<MyClass2> list2 = new ArrayList<>();

        list2.add(new MyClass2("Василий 2"));

        list2.add(new MyClass2("Павел 2"));

        list2.add(new MyClass2("Андрей 2"));

        list2.add(new MyClass2("Петр 2"));

        list2.add(new MyClass2("Анжелика 2"));

        printCollection("Коллекция 2", list2);

    }

    private static void printCollection(String title, List<? super MyClass2> list) {

        System.out.println(title);

        for (Object mc : list) {

            // Т.к. классы - наследники BasicClass, то обе коллекции

            // содержат объекты этого типа.

            BasicClass bc = (BasicClass)mc;

            System.out.println("Item:" + bc.getName());

        }

        System.out.println();

    }

}

run:

Коллеция 1

Item:Василий 1

Item:Павел 1

Item:Андрей 1

Item:Петр 1

Item:Анжелика 1

Коллекция 2

Item:Василий 2

Item:Павел 2

Item:Андрей 2

Item:Петр 2

Item:Анжелика 2

СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)